Программирование

Python

Лекция 6

Кафедра ИВТ и ПМ

2017

План

Outline

Прошлые темы

- Что такое переменная в Python?

Прошлые темы

- Что такое переменная в Python?
- Могут ли две переменные неизменяемого типа ссылаться на одно значение?
- Что будет если это значение будет изменено?
- Назовите неизменяемые типы языка.
- Что происходит при изменении переменной такого типа?
- Что такое сборщик мусора? Как он работает?

Прошлые темы

- Как посмотреть справку по модулю?

Прошлые темы

- Как посмотреть справку по модулю?

  ```
  help ( имя модуля )
  ```

- Назовите некоторые функции пакета random

Прошлые темы

- Как посмотреть справку по модулю?

```
help ( имя модуля )
```

- Назовите некоторые функции пакета random
  - randint

Прошлые темы

- Как посмотреть справку по модулю?

  ```
  help ( имя модуля )
  ```

- Назовите некоторые функции пакета random
  - randint
    randint(a,b) -> псевдослучайное случайное целое от a, до b включительно
  - random

Прошлые темы

- Как посмотреть справку по модулю?

  ```
  help ( имя модуля )
  ```

- Назовите некоторые функции пакета random
  - randint
    randint(a,b) -> псевдослучайное случайное целое от a, до b включительно
  - random
    random() -> псевдослучайное вещественное число от 1 до 0

Прошлые темы

- Как найти максимальное число в списке?

Прошлые темы

- Как найти максимальное число в списке?

```
l = [0, 5, 4, 0, -3, 7]
min = l[0]

for x in l:
    if x < min:
        min = x
```

Если требуется найти индекс минимального числа, вместо отельной переменной для самого числа нужно хранить индекс элемента

Прошлые темы

- Что будет напечатано в результате выполнения этого кода?

```
L = [0]
if L:
    print("OK")
else:
    print("Not OK")
```

Прошлые темы

- Что будет напечатано в результате выполнения этого кода?

```
L = [0]
if L:
    print("OK")
else:
    print("Not OK")
```

OK

- Что будет напечатано в результате выполнения этого кода?

```
L = []
if L:
    print("OK")
else:
    print("Not OK")
```

Прошлые темы

- Что будет напечатано в результате выполнения этого кода?

```
L = [0]
if L:
    print("OK")
else:
    print("Not OK")
```

OK

- Что будет напечатано в результате выполнения этого кода?

```
L = []
if L:
    print("OK")
else:
    print("Not OK")
```

Not OK

Прошлые темы

Что будет напечатано в результате выполнения этого кода?

```
L = 0
if L == [0]:
    print("OK")
else:
    print("Not OK")
```

Прошлые темы

Что будет напечатано в результате выполнения этого кода?

```
L = 0
if L == [0]:
    print("OK")
else:
    print("Not OK")
```

OK

Прошлые темы

Определить, принадлежит ли число x отрезку [-5, 7].
Следующий код содержит ошибку. Какую?

```
if x in (-5,7):
    print("принадлежит")
else:
    print("не принадлежит")
```

Прошлые темы

Определить, принадлежит ли число x отрезку [-5, 7].
Следующий код содержит ошибку. Какую?

```
if x in (-5,7):
    print("принадлежит")
else:
    print("не принадлежит")
```

Проверяется вхождение числа x в кортеж из двух элементов 5 и 7.

Правильный вариант кода?

```
if -5 < x < 7:
    print("принадлежит")
else:
    print("не принадлежит")
```

Прошлые темы

Как разделить строку на слова?
S = "to be or not to be that is the question"

Прошлые темы

Как разделить строку на слова?
S = "to be or not to be that is the question"

```
S.split()
```

['to', 'be', 'or', 'not', 'to', 'be', 'that', 'is', 'the', 'question']

Прошлые темы

Для чего нужен метод **join**?

Что произойдёт в результате выполнения следующего кода?

```
", ".join( [1,2,3,4] )
```

Прошлые темы

Для чего нужен метод **join**?

Что произойдёт в результате выполнения следующего кода?

```
", ".join( [1,2,3,4] )
```

Ошибка связанная с типом:
TypeError: sequence item 0: expected str instance, int found

Как правильно?

Прошлые темы

Для чего нужен метод **join**?

Что произойдёт в результате выполнения следующего кода?

```
", ".join( [1,2,3,4] )
```

Ошибка связанная с типом:
TypeError: sequence item 0: expected str instance, int found

Как правильно?

```
", ".join( ["1", "2", "3", "4"] )
```

1, 2, 3, 4

Прошлые темы

Какая строка будет создана?

```
"{2}. {0} is not {1}".format(True, None, 3.1415)
```

Прошлые темы

Какая строка будет создана?

```
"{2}. {0} is not {1}".format(True, None, 3.1415)
```

```
'3.1415. True is not None'
```

Какая строка будет создана?

```
"{0:5.2f} + {0:12e} + 1 = {1}".format(20.5, "сорок два")
```

Прошлые темы

Какая строка будет создана?

```
"{2}. {0} is not {1}".format(True, None, 3.1415)
```

```
'3.1415. True is not None'
```

Какая строка будет создана?

```
"{0:5.2f} + {0:12e} + 1 = {1}".format(20.5, "сорок два")
```

```
'20.50 + 2.050000e+01 + 1 = сорок два'
```

12 - число позиций для вывода числа.

Вставка юникод символов в строки

- С помощью функции chr

  ```
  S = "why so serious? " + chr(0x263a)
  ```

- С помощью задания кода символа непосредственно в строке

  ```
  S = "why so serious? \u263a"
  ```

- С помощью метода format

  ```
  S = "why so serious? {}".format( chr(0x263a) )"
  ```

why so serious? ☺

Вставка юникод символов в строки

- Код символа задаётся в шестнадцатеричной кодировке
- Перед кодом символа в Unicode таблице должен быть префикс **\u**
- Код символа должен состоять из четырёх чисел.
- Если код не четырёхзначный, то в начало добавляются нули

```
\u0041
```

Замечание о копировании изменяемых объектов

К изменяемым объектам в Python относятся:

Замечание о копировании изменяемых объектов

К изменяемым объектам в Python относятся:

- Списки (list)
- Словари (dict)
- Множества (set)

Замечание о копировании изменяемых объектов

Неправильный способ копирования списков

```
l1 = [1,2,3]
l2 = l1
```

При таком копировании список l2 будет ссылаться на ту же самую коллекцию объектов, что и l1.
Изменении любого из *элементов* списка l2 изменит и список l1.

Замечание о копировании изменяемых объектов

Правильный способ копирования списков
При копировании списка нужно создать копию каждого его элемента.
Например, с помошью среза.

```
l1 = [1,2,3]
l2 = l1[:]
```

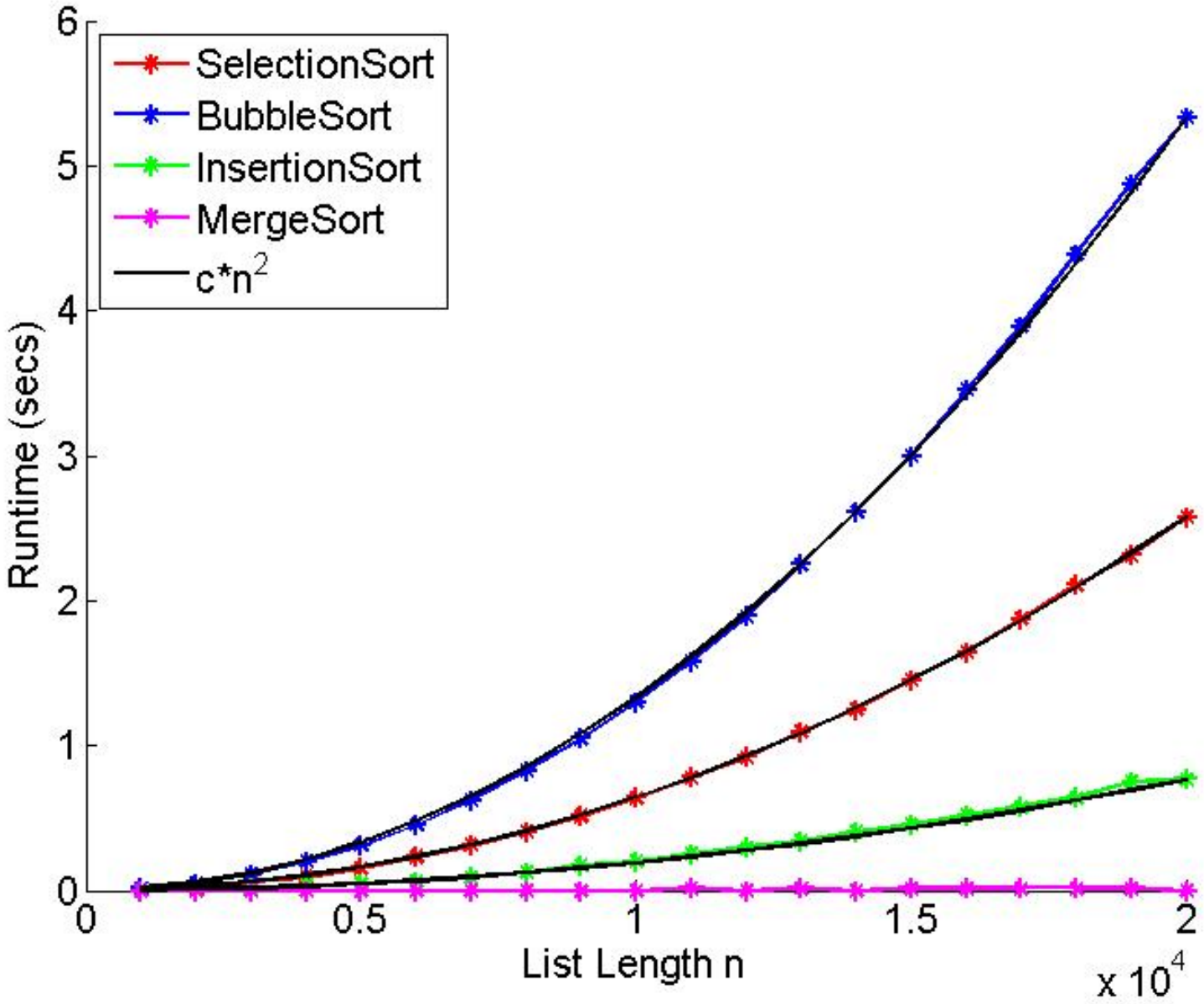
Или с помощью метода копирования copy

```
l1 = [1,2,3]
l2 = l1.copy()
```

При этом стоит помнить, что вложенные списки тоже нужно копировать с помощью вышеописанных способов.

Outline

Алгоритмы сортировки



Outline

Пузырьковая сортировка

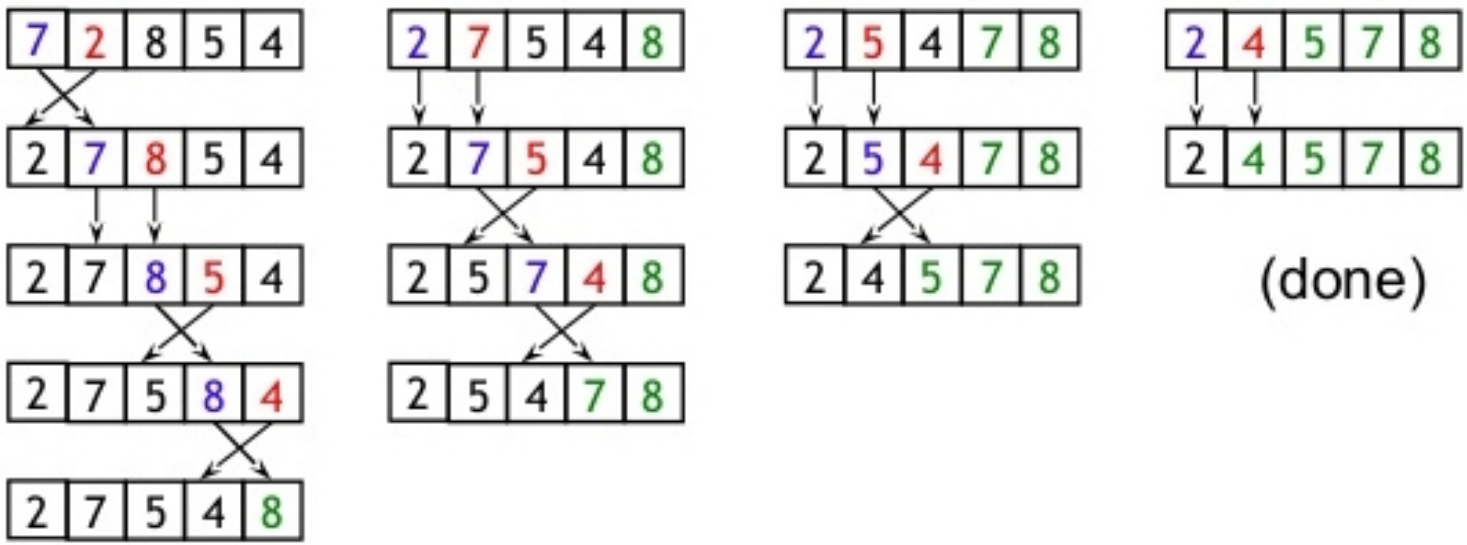
Пузырьковая сортировка (buble sort).

Каждый элемент сравнивается со следующим.
Следующий если следующий элемент больше[1], то элементы меняются местами.
Переходим на один элемент вперёд.
Повторять, пока список не будет отсортирован[2]

[1] предполагается, что сортировка происходит *по возрастанию*

[2] список будет гарантировано отсортирован за известное число шагов

Пузырьковая сортировка



1. После первого прохода наибольший элемент оказывается на свей позиции.
2. После второго прохода второй по величине элемент тоже стаёт в свою позицию.
3. И т.д.

Пузырьковая сортировка

- **Наихудший случай**: список отсортирован по убыванию.
  Прежде чем максимальный элемент попадёт на свою позицию, его нужно будет поменять местами со всеми остальными элементами. Для второго по величине элемента - число обменов меньше на 1, и т. д.
- **Наилучший случай**: список отсортирован по возрастанию.
  Переставлять элементы не нужно.
  Алгоритм пройдёт список такое же число раз, как и в худшем случае.

Время работы алгоритма пузырьковой сортировки пропорционально n^2
где n - длинна списка.

Outline

Сортировка выбором

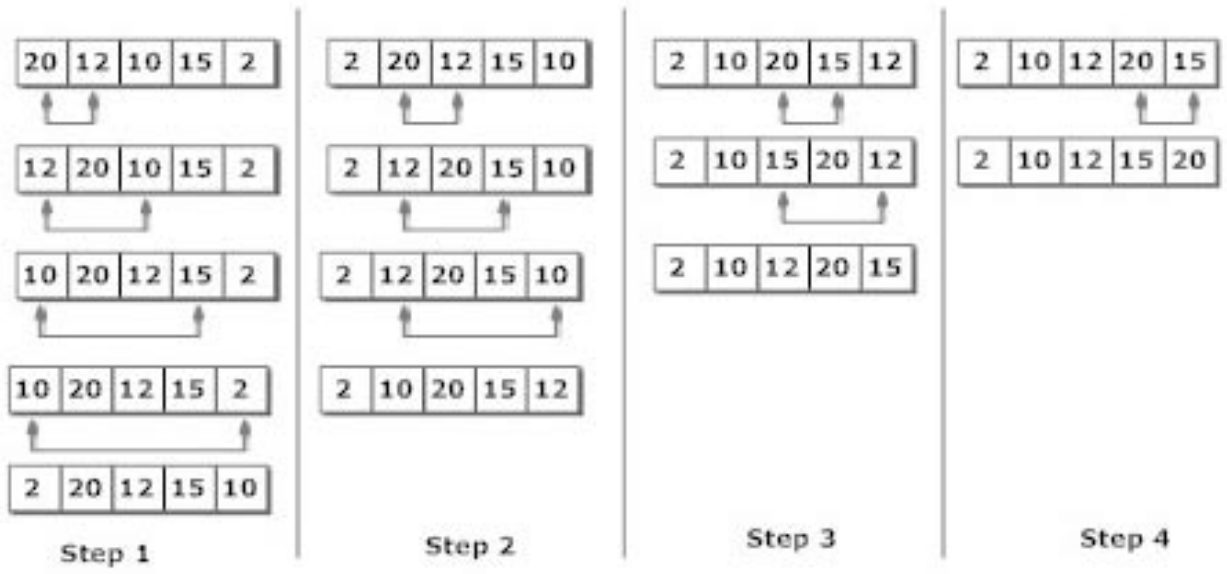


Figure: Selection Sort

Сортировка выбором

1. находим номер минимального значения в текущем списке
2. производим обмен этого значения со значением первой не отсортированной позиции (обмен не нужен, если минимальный элемент уже находится на данной позиции)
3. теперь сортируем хвост списка, исключив из рассмотрения уже отсортированные элементы

Сортировка выбором

- **Наихудший случай**: список отсортирован по убыванию.
  Нужно переставить каждый элемент в свою позицию.
- **Наилучший случай**: список отсортирован по возрастанию.
  Переставлять элементы не нужно.
  Алгоритм пройдёт список такое же число раз, как и в худшем случае.

Время работы алгоритма пузырьковой сортировки пропорционально n^2
где n - длинна списка.

Outline

Измерение времени работы кода в Jupyter

```
%time
```

измеряет время работы одной строчки кода

```
sum = 0
%time for i in range(10000000): sum += i

CPU times: user 1.82 s, sys: 4 ms, total: 1.82 s
Wall time: 1.83 s
```

Результат:1.83 секунды
Измерить время работы отдельной строчки кода в Jupyter
Лучше всего отдельные части программы помещать в функции(подпрограммы) для измерения времени их работы

Измерение времени работы кода в Jupyter

Лучше всего отдельные части программы помещать в функции(подпрограммы) для измерения времени их работы

```
def my_very_fast_code():
    sum = 0
    for i in range(10000000):
        sum += i
        sum =+ = sin(sum)
```

```
%time my_very_fast_code()
```

CPU times: user 2.94 s, sys: 0 ns, total: 2.94 s
Wall time: **2.94 s**

Измерение времени работы кода в Jupyter

Однако такое измерение для кода, работающего меньше минуты может быть не точным.
Это связано с тем, что в это же время процессор может быть загружен другой программой

Поэтому в Jupyter для коротких вычислений лучше использовать команду **%timeit**

Эта команда запускает код несколько ра и вычисляет среднее время его работы.

Измерение времени работы кода в Jupyter

```
def my_very_fast_code():
    sum = 0
    for i in range(10000000):
        sum += i
        sum += sin(sum)
```

```
%timeit my_very_fast_code()
```

3.05 s ± 147 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)

Среднее время работы: 3.05 секунды

Измерение времени работы

Измерить время выполнения кода можно и не прибегая к средствам Jupyter. Для этого понадобится модуль для работы

со временем **time**

функция **time** одноимённого модуля возвращает вещественное число - текущее время в секундах

Измерение времени работы

Тогда измерение времени работы кода можно организовать следующим образом

```
from time import time

t0 = time()
... my code ...
dt = time() - t0

print(dt)
```

Время работы кода будет записано в переменной dt.

Ссылки и литература

- пузырьковая сортировка: танец
- пузырьковая сортировка: анимация

Ссылки и литература

Ссылка на слайды

github.com/VetrovSV/Programming